(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 January 2002 (24.01.2002)

PCT

(10) International Publication Number
WO 02/06918 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number: PCT/US01/18333

(22) International Filing Date: 6 June 2001 (06.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/616,699 14 July 2000 (14.07.2000) US

(71) Applicant: TELCORDIA TECHNOLOGIES, INC. [US/US]; 445 South Street, Morristown, NJ 07960-6438 (US).

(72) Inventors: LIU, Changdong; 27 Lawton Road, Bridgewater, NJ 08807 (US). RAMAMURTHY, Suryanarayan; 17 Victoria Drive, Eatontown, NJ 07724 (US). WEI, John; 667 Lloyd Road, Aberdeen, NJ 07747 (US).

(74) Agents: GIORDANO, Joseph et al.; c/o International Coordinator, Telcordia Technologies, Inc., Room 1G112R, 445 South Street, Morristown, NJ 07960-6438 (US).

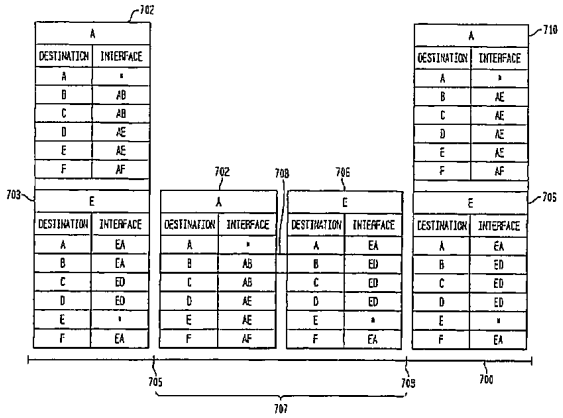
(81) Designated States *(national)*: CA, JP.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A METHOD, SYSTEM, AND PRODUCT FOR PREVENTING DATA LOSS AND FORWARDING LOOPS WHEN CONDUCTING A SCHEDULED CHANGE TO THE TOPOLOGY OF A LINK-STATE ROUTING PROTOCOL NETWORK



(57) Abstract: In the context of network topology changes to Internet Protocol networks, an approach is disclosed which eliminate pthologies associated with the removal of links and routers from the network. Specifically disclosed are several techniques which prevent the loss of data packets across failed links during the transient period following the deletion of a link from the ntwork topology and which prevent the formation of forwarding loops which would otherwise follow such a link deletion during a convergence period in which adjacent routers are in conflict as to the shortest path for forwarding of data packets. The techniques used to achieve these improvements include preparing a network in advance for a secheduled network change in order to avoid data loss, and conducting updates, also prior to the scheduled network change, to the routing tables of affected routers in a specific order to prevent forwarding loops. The present invention may take the form of a centralized implementation or a distributed implementation, or a combination of both.

WO 02/06918 A2

A METHOD, SYSTEM, AND PRODUCT FOR PREVENTING DATA LOSS AND FORWARDING LOOPS WHEN CONDUCTING A SCHEDULED CHANGE TO THE TOPOLOGY OF A LINK-STATE ROUTING PROTOCOL NETWORK

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates generally to networks and more particularly, to a method, system, and product for preventing data loss and forwarding loops when conducting a scheduled change to the topology of a link-state routing protocol network. Preferably, the network is Internet Protocol ("IP")-based and the link state routing protocol employed within the IP based network is the Open Shortest Path First ("OSPF") routing protocol.

Brief Description of the Prior Art

A fundamental part of network computing is seamless data transfer. Networks connect a plurality of computers situated in various locations through out the world and transport information to these locations. Networks, which are based on the Internet Protocol ("IP"), are rapidly becoming the norm.

IP based networks use a number of different IP routing protocols, including, for example, Routing Information Protocol (RIP; cf. request for comments (RFC) 1053), Open Shortest Path First (OSPF; RFC 1583), Intermediate System-to-Intermediate System (IS-IS; ISO 10589), Distance-Vector Multicast Routing Protocol (DVMRP; RFC 1075), and Border

Gateway Protocol (BGP; RFC 1771). Each of these protocols determine how packets ought to be routed throughout the routing domain of the network.

Currently, two routing protocols predominate - a distance vector routing protocol and a link-state routing protocol. In the distance vector protocol, the routers cooperate in performing an incremental, distributed computation. Each router advertises a list of distances separately, usually trying to find paths that minimize a simple metric like the number of hops to the destination. An example of a distance vector routing protocol is the Routing Information Protocol, or RIP.

The link-state routing protocol, on the other hand, employs a replicated, distributed network topology database approach. Each router contributes pieces to this database by describing the router's local environment (the link's state). This description or link state may contain, for example, a list of the router's interfaces which participate in active communication between the router and local IP network links, the neighboring routers with which the router may directly communicate through the interfaces and across the links, and a parameter called the cost or metric assigned to each of the links. Generally, each router in the link-state routing protocol is capable of calculating its own routing table based on its copy of the network topology database. An example of a link-state routing protocol is the Open Shortest Path First routing protocol or OSPF.

A complex Internet Protocol ("IP") network typically consists of multiple sub-networks. The multiple sub-networks are connected to one another via routers and communication links. IP networks are designed to facilitate the transmission of data packets, or IP packets, from one sub-network to another. An IP packet bound for a particular

destination within a sub- network will contain, as part of its header, information which identifies that particular destination as the destination of the IP packet.

Routers in an IP network will normally have at least one active interface through which communication will be made possible. To reach its destination, an IP packet will cross one or more communication links. A communication link is typically interposed between two given routers in an IP network. The link can either be unidirectional or bi-directional. In an IP network, a routing table at a router instructs the router how to forward data packets throughout the network. The routing table is typically constructed according to a predefined routing protocol. That is, the routing table indicates which links a particular data packet would need to traverse in order to reach its desired destination. Each individual link along the destination route is chosen based on whether it is a best (shortest) path towards the data packet's final destination.

For example, the well-known OSPF is a dynamic link-state routing protocol whereby data packets are transmitted from source to destination within a network according to the shortest path to all known destinations within the network. OSPF is an Internet Engineering Task Force (IETF) recommended standard that has been extensively deployed and exercised in many networks.

OSPF uses a complicated link-state algorithm to build and calculate the shortest path to all known destinations within the network. What follows is a simplified description of the various steps involved:

1) During initialization a router within a network routing domain will generate a link-state advertisement ("LSA"). An LSA describes the local state of a particular router or network including the state of the router's interfaces.

2) The LSA is flooded throughout the routing domain of the network. The collected link state advertisements of all routers and networks form the protocol's network topological database.

3) Next, the router will calculate a Shortest Path Tree to all destinations reachable within the network. The router uses the well-known Dijkstra algorithm to calculate the shortest path tree. The destinations, the associated cost and the next hop to reach those destinations will form the router's routing table. In operation, the Dijsktra algorithm places the router at the root of a tree and calculates the shortest path to each destination based on the cumulative cost required to reach that destination. Each router within the network would perform this process. Hence, each router will have its own view of the topology even though all the routers will build a shortest path tree using the same link-state database.

Typically, the cost or metric of an interface corresponds to the overhead required to send data packets across the interface. This cost is inversely proportional to the bandwidth of the interface, for example, a higher bandwidth indicates a lower cost. Therefore, it is more costly to cross a 56k serial line than to cross a 10M Ethernet line. The following formula is used to calculate the cost of an interface:

$$\text{COST} = \frac{10 \text{ EXP } 8}{\text{BANDWIDTH IN BPS}}$$

So, for example, to cross a 10M Ethernet line it will cost 10 (10 EXP8/10 EXP7 = 10). While the default cost of an interface is calculated based on the bandwidth, the cost of an interface can nonetheless be forced or arbitrarily set as the market may allow or dictate.

Referring to Figure 1A, assume we have a network diagram with the indicated interface costs. In order to build the shortest path tree for router A, router A is positioned at the root of the tree. Next the lowest cost to each destination within the network is calculated. Router A can reach network 192.213.11.0 via router B with a cost of 15 (10+5). Router A can also reach network 222.211.10.0 via router C with a cost of 20 (10+10) or via router B with a cost of 20 (10+5+5). Where equal cost paths exist to the same destination, some OSPF implementations will keep track of up to a certain number of next hops to the same destination. After the router builds the shortest path tree, it will start building the routing table accordingly. Directly connected networks will be reached via a metric or cost of 0 and other networks will be reached according to the cost calculated in the tree. As shown in Figure 1A, the cost for router A to reach network 128.213.0.0 is 0 because router A is directly connected to network 128.213.0.0.

When there is a change to the topology of a network, i.e., the deletion of a network resource such as a communication link, use of the link state routing protocol may result in data loss and the formation of transient (i.e., temporary) forwarding loops. For example, in response to a network topology change, each router would update its routing information and re-calculate shortest paths between the router and the destination station. This newly updated information is then transmitted to the other routers in the network. While the other routers propagate the updated information throughout the network and re-calculate their forwarding tables accordingly, routers possessing inconsistent routing information, that is, routers that have not yet updated their routing tables to account for the topology change, may forward

packets back to a previous sender of the packets, thereby creating forwarding loops and the router or routers adjacent to the failed link may continue to forward packets across the failed link, thereby creating data loss. The process of getting routing tables at all routers within a given routing domain or network synchronized after a network change is called convergence.

Conventional link state routing protocols, such as OSPF, do not prevent data loss or suppress the formation of forwarding loops during this period of convergence. While OSPF was developed to quickly detect topological changes in a network (such as router interface failures) and calculate new loop-free routes after a short period of convergence, nevertheless, forwarding loops could still be created and data could still be lost during the short period of convergence.

Where a change to the network topology is anticipated, however, data loss and forwarding loops can be eliminated. Accordingly, the present invention is directed to a novel and inventive approach to preventing data loss and forwarding loops during scheduled network topology changes. Two critical features of the present invention include: 1) updating the routing tables of all network routers which will be affected by the link removal prior to the time of the change, rather than after the change, so as to avoid data loss during the period just after the change, and 2) conducting those routing table updates in a particular sequence such that the updates of all routers farther away from the change point (upstream routers) occur before routers nearer to the change point (downstream routers), so as to prevent forwarding loops during the convergence period.

SUMMARY OF THE INVENTION

In accordance with one aspect of the present invention, there is provided in a link state routing protocol network including a plurality of routers having a routing table associated with each of the router and a plurality of links, a method for preventing data loss and forwarding loops during a scheduled network topology change, comprising the ordered steps of first identifying all affected routers to be changed to effectuate the scheduled network topology change, second, performing a predetermined routing table updating sequence for the scheduled network topology change and third, performing the scheduled network topology change.

In accordance with a second aspect of the present invention, there is provided, in a link state routing protocol network including a plurality of routers having a routing table associated with each of the router and a plurality of links, a computerized system for preventing data loss and forwarding loops during a scheduled network topology change, comprising: a central processor; and a program module associated with the central processor for 1) identifying all affected routers to be changed to effectuate the scheduled network topology change; 2) performing a predetermined routing table updating sequence for the scheduled network topology change; and 3) performing the scheduled network topology change.

In accordance with a third aspect of the present invention, there is provided a computer program product for preventing data loss and forwarding loops during a scheduled network topology change to a link state routing protocol network, the network having a plurality of routers having a routing table associated with each of the router and a plurality of links, the computer program product comprising a computer usable medium having computer readable program code embodied in the medium, the program code including: computer

readable program code embodied in the computer usable medium for causing a computer to 1) identify all affected routers to be changed to effectuate the scheduled network topology change; 2) perform a predetermined routing table updating sequence for the scheduled network topology change; and 3) perform the scheduled network topology change.

In accordance with a fourth aspect of the present invention, there is provided In a link state routing protocol network having a network manager, the network including a plurality of routers having a routing table associated with each of the router and a plurality of links, a method, to be performed by the network manager, for preventing data loss and forwarding loops during a scheduled network topology change, comprising the ordered steps of: identifying all affected routers to be changed to effectuate the network topology change; performing a predetermined routing table updating sequence; and performing the scheduled network topology change.

In accordance with a sixth aspect of the present invention, there is provided, in a link state routing protocol network including a plurality of routers having a routing table associated with each of the router and a plurality of links, a distributed method for preventing data loss and forwarding loops during a scheduled network topology change, comprising the steps of: identifying and generating a list of affected routers to be changed to effectuate the network topology change; generating a plurality of link state forecast messages, each link state forecast message containing the list of affected routers and a forecast of the network topology change; forwarding the link state forecast messages to each of the affected routers to be acknowledged by each of the affected routers; generating a post change routing table; updating the routing table of each of the affected routers in accordance with the post change routing table when all outstanding link state forecast messages have been acknowledged; and performing the scheduled network topology change.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features, aspects, and advantages of the present invention will become better understood with regard to the following description, appended claims, and accompanying where:

Figure 1A shows an example of a network having a plurality of network routers, A-D, and a plurality of links, each link bearing it's associated cost. Figure 1A also shows a shortest path tree (SPT) of router A spanning all of the network routers.

Figure 1B shows an example of a shortest path tree (SPT) of router 101 spanning all network routers of a hypothetical network.

Figure 2 shows an example of an reverse shortest path tree (RSPT) of router 101 spanning all network routers of the hypothetical network of Figure 1B.

Figure 3 shows an example of a critical RSPT sub tree of the router 101 of Figure 1B in the hypothetical network of Figure 1B.

Figure 4 shows one embodiment of a typical IP network having routers A-F and a plurality of link pairs.

Figure 5 shows the network of Figure 4, during the convergence period, forming a transient forwarding loop after the failure of link AB.

Figure 6 shows the contents of the routing tables of router A and router E before, during, and after the convergence period following the failure of the link AB within the network of Figure 4.

Figure 7 shows the RSPT of router B spanning all network routers within the network of Figure 4.

Figure 8 shows the critical sub-tree of the RSPT of router B comprising all affected routers within the network of Figure 4.

Figure 9 shows the contents of the routing tables of router A and router E before, during, and after the convergence period after implementation of the present invention within the network of Figure 4.

Figure 10 is one preferred implementation of the present invention utilizing a centralized approach.

Figure 11 is another preferred implementation of the present invention utilizing a distributed approach.

DETAILED DESCRIPTION OF THE INVENTION

Referring more specifically to the drawings, for illustrative purposes the present invention is embodied in the system configuration, method of operation and product generally shown in Figures 1 – 11. It will be appreciated that the system, method of operation and program product may vary as to the details of its configuration and operation without departing from the basic concepts as disclosed herein.

Figure 1B is an example of an SPT of router **101** spanning all network routers of a hypothetical IP network **100**. A root router **101**, designated with the letter "R", appears near the center of the diagram, and the shortest paths available for transmission of IP packets from the root router to all other network routers **102-116** are shown across unidirectional links **117-131**. For the purposes of this disclosure, network routers **102-116** are designated by their "level" within the SPT with Roman numerals I, II, III and IV corresponding to the number of links or "hops" across which IP packets are transmitted on the path from the root router to each network router. Hence, a Level I router is one hop away from the root, a Level II router is two hops away from the root and so on.

Figure 2 is an example of an RSPT of router **101** spanning all network routers of the network **100** of Figure 1. The identity and location of network routers **101-116** are also the same in Figure 2 as they are in Figure 1. Network routers **102-116** are, as they were in Figure 1, also similarly designated according to their level. The direction of IP packet transmission in Figure 2, however, is the reverse of the SPT illustrated in Figure 1. By the word "reverse" is meant that the paths shown in Figure 2 are the shortest paths available for transmission of

IP packets to (rather than from) the root router **101** from (rather than to) all other network routers **102-116**. The reverse shortest paths shown utilize uni-directional links **417-426** (which are the pairs of unidirectional links **117-126**), **428** (the pair of unidirectional link **128**), **431** (the pair of unidirectional link **131**), and **432-434** (which do not correspond to any unidirectional link shown in Figure 1).

The reverse shortest path routing requirement is implemented by certain portions of the data contained within the routing tables resident on each network router **102-116**. The RSPT shown in Figure 2 does not represent an arrangement of data which is required by standard operation of a link-state routing protocol to be stored at the root router **101**, as is the case in the SPT of Figure 1, but is instead a representation of the network organization which is implemented, in concert, by all routers that forward IP packets to the root router **101**, according to the link state routing protocol in use.

It is noted here, that it is generally assumed that the RSPT and SPT for root router **101** within network **100** would differ only in the direction of IP packet transmission. However, the existence of asymmetric link metrics can produce the result in which the collection of links and routers which comprise the shortest path from a given router A to a given router B would be different from the collection of links and routers which comprise the shortest path from router B to router A. As a comparison of the two Figures 1B & 2 illustrates, the difference may involve not only the identity of the links, but also the total number of links.

To elaborate on this point further, the differences in the identity and number of shortest path links of a RSPT as compared to a SPT is demonstrated with reference to network router **115**. In Figure 1B, router **115** is an SPT level IV router. In Figure 2, router **115** is an

RSPT level III router. Accordingly, fewer "hops" are needed for the reverse shortest path and different intermediate links and routers participate depending on the direction of IP packet transmission. In Figure 2, network routers **112**, **114** and **116** maintain their designation as level II, III and IV routers, respectively, as in Figure 1B. However, they too illustrate the enforced use of a different set of intermediate links and routers depending on the direction of IP packet transmission.

Root router **101** does not store any information regarding it's RSPT. It's RSPT must be calculated when needed. A network router may calculate its RSPT using the well-known Dijkstra's shortest path algorithm by taking from its local topology database link metrics in the direction from (rather than to) all other routers. Thus, all minimally functional routers in IP networks today are capable of performing this calculation without the addition of new functionality.

Referring again to Figure 2, the RSPT for a given router in a network will typically be made up of multiple "sub-trees" which appear to emerge from the router at distinct input interfaces of the router where the router is located at the root of the RSPT. However, it is not necessary to consider the configuration of each sub-tree of a relevant RSPT. In fact, the only RSPT sub-trees which require consideration in a given relevant RSPT are those which include the component (e.g. link or router) that is to be removed from the network. These sub-trees are deemed "critical" RSPT sub-trees. Thus, if unidirectional link **417**, as shown in Figure 2, were scheduled for removal from network **100**, the sub-tree containing it, as shown in Figure 3, would qualify as a critical RSPT sub-tree.

With the exception of the root router, all routers appearing within a critical RSPT sub-tree are deemed "affected" routers. Every affected router may not need to update its routing table in response to a scheduled network change. However, regardless of whether the routing tables of all of the affected routers is required to update in anticipation of the removal of link **417**, it is essential for the purposes of this invention that routers **102, 107, 108,** and **115** all be considered affected routers and that they all participate in the routing table updating sequence described herein.

Figure 4 shows one embodiment of a typical IP network having 5 routers, routers A-F and 12 unidirectional links or 6 link pairs, **207-218**. Figure 5 shows the network of Figure 4, during the convergence period, forming a transient forwarding loop after the failure of link pair **207/208**. Figure 6 shows the contents of the routing tables of router A and router E before, during, and after the convergence period following the failure of the link **207** within the network of Figure 4. Timeline **300** begins at event **301**, which represents the failure of unidirectional link **207**, or link **"AB"**. Immediately after event **301**, the contents of the routing tables of routers A and E remain as they were prior to event **301**, and they are shown in Figure 6 as routing tables **302** and **303** respectively. The first action taken by router A once it has detected the failure of link **AB(207)** is to send an LSA to its neighbors, routers E and F, containing a notification of the change in network topology that the failure of link **AB(207)** represents. The time span **304** during which this action is taken is shown in Figure 6. Also during the timespan **304**, router A calculates an updated routing table, and compares the updated routing table to its existing routing table to determine which entries in the two tables are dissimilar. Since the forwarding interface for the best path to reach router B from router A will have changed from link AB to link AE, router A will update its routing table to reflect the new best path, which it does at event **305**. Until this is done, router A could still forward

data to router B over the already failed link **AB(207)**. If this occurs, the forwarded data would be lost.

The concept of forwarding loops may be understood with regard to the events that follow event **305**. As did router A, router E is also supposed to update its routing table, after receipt of the LSA from router A and in the same fashion as router A, performed its routing table update. Router E's routing table update will change the forwarding interface for the best path to reach router B from link EA to link ED. However, there is a lag between these two routing table updates, as is evident on the timeline between event **305**, when the routing table of router A is updated, and event **309**, when the routing table of router E is updated. This lag, during which time router E also forwards an LSA to router D, is known as the convergence period **307** relevant to routers A and E. During the convergence period **307**, the updated routing table **306** at router A, and the unchanged routing table at router E **303** demonstrate an inconsistency **308** regarding the best path to router B. A forwarding loop **219** forms as a result of this inconsistency and persists until event **309**, after which the updated routing tables **306** and **310** at routers A and E respectively, are again in harmony as to the best path to reach router B. In this example, because link **208** failed along with **207**, routers B and C are also subject to this pathology. The same problem may occur, but with a much smaller probability, when a new link is brought up or added to the network (not shown in Figures 2 or 3).

Consider the scenario where Link pair **207/208** is scheduled for deletion. As such, Routers A and B, being adjacent to the link to be removed, are the relevant routers whose RSPT's must be generated. For the purposes of simplicity, the following description will only focus on what takes place with router B, bearing in mind that the same steps will also be performed with respect to router A.

In operation, the basic invention works as follows: at step 1, a RSPT of router B is generated. Figure 7 shows the RSPT of router B spanning all network routers of the network of Figure 2. Here, Routers A and C are Level I routers, and Routers D, E and F are Level II routers. At step 2, a critical RSPT sub-tree of router B is generated by performing a breadth first search on the RSPT sub-tree of Router B that starts with router B and continues along the Link BA branch out towards the leaf routers. Figure 8 shows this critical RSPT sub-tree. All routers comprising this sub-tree are affected routers, that is, routers that require updating in response to the deletion of Link AB. At step 3, an updating sequence is obtained, by sorting the routers comprising the critical sub-tree in descending order based on the vertex labeling (the routers Level designation) generated during the breadth-first search. The basic rule in defining an updating sequence is that any affected router cannot update its routing table until all its descendants have done so. However, among the affected routers of a same generation or Level, no particular sequence needs to be enforced.

As shown in Figure 7, Routers E and F are also leaf routers in the critical RSPT sub-tree of router B. Accordingly, the routing table of router E and router F will be among the group of routing tables to be updated first. To again simplify the discussion, the focus will be on router E as the same steps are preformed with respect to router F. The routing table of router A will be updated last among affected routers because router A is a level I router in the critical RSPT sub-tree of router B. Figure 9 shows the contents of the routing tables of routers A and E, before, during, and after the convergence period in network **200** in preparation for the scheduled removal of link **AB(207)**. Up to time 705, prior to the scheduled removal of the link, the routing tables of routers A and E are in harmony as to the shortest path for forwarding IP packets to router B. At time 705, the routing table for router E is updated to account for the scheduled removal of link **AB(207)**. At time 709, the routing

table of router A is updated. However, in between the foregoing two updates, a time lag, or convergence period **707** exists. During this convergence period, the updated routing table of router E, and the unchanged routing table of router A, are inconsistent with regard to the best path to router B. The inconsistency 708 persists until time 709, when the routing table for router A is updated. However, this inconsistency is not of the sort by which a forwarding loop between routers A and E may develop. Now the updated routing tables of routers A and E, are again in harmony as to the best or shortest path to reach router B. Now link **207(AB)** can successfully be removed without triggering forwarding loops between routers A and E since neither router will rely on the presence of link **207(AB)** within the network to complete the shortest path to router B or any other network router.

During the execution of the method of the present invention the distributed link-state database remains unaltered, that is it remains consistent with the real network topology at all times. Thus, after the scheduled change is performed the distributed link-state database within the network (which has not yet accounted for the change) will require updating to bring it in line with the true network topology (which includes the change). This necessary update can take place via conventional means, such as by means of LSAs which are automatically generated by routers adjacent to the change point in response to such a change. However, when the LSA-driven topology database update is performed within a network, which had been prepared in advance for the change by the use of the method of the present invention, no change to any routing table in use within the network will be performed. This is because all affected routers which would have required a routing table update in response to the change will already have received such an update by the time the LSA is generated.

Figure 10 is a process flow diagram setting forth one preferred implementation of the present invention wherein a centralized managing entity or manager performs all necessary preparatory and operational steps of the present invention, external to the network. This manager can be a person, or a managing computer resource, such as the widely used Simple Network Management Protocol ("SNMP") manager. SNMP is an Internet standard protocol, defined in STD 15 RFC 1157, that was developed to manage network devices such as routers. The following description focuses on how the SNMP manager implements the present invention. However, as indicated earlier, the invention is not limited thereby as a "person" can fully perform the operational steps of the present invention.

First, the SNMP manager will obtain a copy of an accurate distributed network topology database. This is done according to conventional means. Next, the SNMP manager will prepare the network for the scheduled topology change. Assuming that the scheduled network change is to take down the link pair between routers A and B (one from A to B, the other from B to A), the SNMP manager will identify all affected routers and determine and control the two updating sequences corresponding to the two critical RSPT sub-trees with each RSPT sub-tree having either router A or B at the root. Coordination between the two sequences is not required.

With regard to each individual critical RSPT sub-tree, the SNMP manager will proceed independently. Going router by router and for each router in the critical RSPT sub-tree at issue, starting from the leaf routers of the various shortest paths, and proceeding to the level I router, at step **1000**, the SNMP manager will first compute on the router's behalf the routing table as if link L had been taken down (the "post-change" routing table). At step **1005**, the SNMP manager will compare the corresponding entries of the post-change routing

table with the current routing table of the router (the SNMP manager may poll the router for a copy of its current routing table or compute it by itself). At step **1010**, where differences appear, the SNMP manager will use the SNMP "SetRequest" command to update the router's routing table in accordance with the post-change version. At step **1015**, the SNMP manager would then continue to the next router in the sequence, and the next router, etc., until it had finally performed the necessary update to the routing table of the RSPT level I router associated with the particular critical RSPT sub-tree.

When execution of both updating sequences finishes, there would no longer be any data flows crossing the link pair, and as such, the link pair may safely be taken down. While taking down the link pair will cause routers A and B to originate new LSAs to announce the change, the routers receiving the LSAs will only perform topology database update in accordance to the LSA. Specifically, the LSA will not cause any change to the routing table of a particular router receiving the LSA because the routing table of the router will not depend on the existence of the link pair to complete any shortest paths within the network.

In another preferred implementation, rather than utilize an external managing entity to perform the method of the present invention, the method may alternatively be implemented in a distributed fashion. Figure 11 is a process diagram showing this second implementation of the present invention. As described before, the SNMP manager must compare the current routing table with the post-change routing table for every affected router. While computing new routing tables in reaction to network changes is inevitable, it is indeed introducing a lot of overhead for the SNMP manager to acquire a copy of the current routing table of each affected router, regardless of the way it is acquired. If the SNMP manager computes a copy by itself, the overall computational cost roughly doubles because the concerned router has

done the same calculation. If retrieving the copy from the concerned router, the SNMP manager may retrieve it on an on-demand basis, or pre-retrieve it and store it for later use. On-demand retrieving creates lots of control traffic (nowadays, typical full routing tables in the Internet consist of 45,000 – 60,000 entries), also prolonging the entire updating procedure. Pre-retrieving implies that the SNMP manager must maintain a copy of the current routing table for each and every router, which involves an original retrieve at the setup time and an incremental retrieve after each updating.

Because in the link state routing domain, every router maintains an accurate replicated topology database, the distributed version of the present invention can be more efficient than the SNMP manager approach. From the descriptions below, it will be shown that a correct updating sequence is inherent to the distributed version of the present invention. Therefore, there is no need for explicit computation of such a sequence.

In the distributed approach, the routers adjacent to the site of the scheduled network topology change are notified of the time and nature of the upcoming change. Assuming the network change is to take down the link pair between routers A and B (Link AB and Link BA, alternatively referred to in combination as Link L), router A and router B will both be made aware of the scheduled deletion of the link pair.

Router A and router B will thereafter perform the method of the present invention separately. Again, no coordination between the two procedures is required. For illustration purposes, what follows is a description of how router A, as a root router, performs the steps of the distributed approach. Router B will perform the same steps.

Referring to Figure 11, at step 1100, router A identifies all affected routers associated with the scheduled removal of link BA.

At step 1105, router A generates a specialized message called a link state forecast message (hereinafter referred to as an LSF message). Here, router A need not sort and sequence the list of affected routers according to level as was described previously, for example, in the SNMP manager approach, router A need only generate and send an LSF message. The LSF contains two critical components: 1) the list N_a^i of affected routers and 2) information forecasting the future event that link BA will be taken down. Unlike an LSA, which is flooded throughout the entire IP network routing domain, an LSF targets only affected routers. An LSF is similar, however, to an LSA in that they have to be acknowledged by the receiving affected router. This process is the same as the conventional LSA acknowledging mechanism. The LSF aids in the distributed approach by using the inherent organization of the network topology to enforce the routing table updating sequence critical to the method of the present invention.

At step **1110**, router A sends the LSF through it's interface to link L, IF_L. At step **1115**, the LSF is received by router B. (Note that the following description applies not only to router B, but to any router which receives an LSF). Router B, upon receiving an LSF, checks to see if it is in the updating list N_a^i. Because this is true for router B, router B will proceed. In the case of any receiving router for which this is not true, the LSF will be discarded.

At step **1120**, router B partitions the updating list N_a^i based on its forwarding interfaces according to the following: $N_a^i = \{N_a^{i+1}(IF_0), N_a^{i+1}(IF_1), N_a^{i+1}(IF_2), \ldots, N_a^{i+1}(IF_k)\}$,

in which: *forwardIf* (R) = IF_j iff $R \in N_a^{i+1}(IF_j)$, $j \in \{0,1,2,\ldots, k\}$, and where *forwardIf()* is the function that returns the forwarding interface for each given destination. Here, it is assumed that the link metrics are symmetric for clean presentation. When link metrics are asymmetric, many conventional techniques can be used to identify the proper interfaces. When equal cost multipaths exist, any tiebreak algorithm may be used to make this true: $N_a^{i+1}(IF_m) \cap N_a^{i+1}(IF_n) = \varnothing$, $m,n \in \{0,1,2,\ldots, k\}$, $m \neq n$ }, (think that *forwardIf(this_router)* = IF_0, the local loop-back interface).

At step **1125**, if the receiving router (in this case, router B) is a leaf router; that is, if $N_a^{i+1}(IF_j) = \varnothing$, $j \in \{1,2,\ldots,k\}$, the leaf router will compute a post-change routing table, compare it to the existing routing table of the leaf router, and if necessary, update the leaf router's routing table. Thereafter, the leaf router generates an acknowledgement of $LSF(N_a^i)$ and sends the acknowledgement to the router which sent the $LSF(N_a^i)$ to the leaf router.

At step **1130**, if the receiving router (in this case, router B) is not a leaf router; that is, if $N_a^{i+1}(IF_j) \neq \varnothing$, $j \in \{1,2,\ldots,k\}$, the receiving router must present $LSF[N_a^{i+1}(IF_j)]$ (obtained by replacing N_a^i in $LSF(N_a^i)$ with $N_a^{i+1}(IF_j)$) at interface IF_j. The receiving router then computes a post-change routing table. Any update to the router's routing table which may appear necessary upon comparison of the post-change routing table to the existing routing table must not take place until all outstanding LSFs are acknowledged. Once this is true, the router must, if such update proves necessary, update the router's routing table. Thereafter, the router must generate an acknowledgement of $LSF(N_a^i)$ and send that acknowledgement to the router which sent $LSF(N_a^i)$ to the router.

When both updating sequences are finished, data will no longer flow on either link AB

or link BA. As such, the link pair L may safely be taken down. In taking down the link, routers A and B will originate LSAs to announce the change. However, the LSA will not cause any change to the routing table of a particular router receiving the LSA because the routing table of the router will not depend on the existence of link L to complete any shortest paths within the network.

An advantage of the present invention is that because it preserves the accuracy of the link-state database throughout its execution, routers within a network executing the present invention will remain as responsive to unexpected failures within the network as they would be the absence of the execution. Thus, should an unexpected failure occur while a session of the present invention is executing, the session shall be preempted and will terminate immediately. Furthermore, once the network has reacted to the unexpected change and is once again stable, the previously terminated session will restart from the beginning, subject to any further unexpected failures, which may thereafter occur.

It is anticipated, however, that regular use of the present invention method to facilitate maintenance and reconfiguration tasks within networks will reduce the frequency and likelihood of unexpected failures of network resources. One reason for this effect is the fact that the present invention can be performed in a network during periods characterized by a high volume of data traffic without risk of disruption or loss of data. Thus operators need not seek to delay the performance of urgently needed maintenance or reconfiguration tasks such that they take place during periods characterized by a low volume of traffic, e.g., between midnight and six in the morning, depending on the type of network. Another reason is that the availability of the present invention method makes more frequent maintenance of network resources possible by significantly reducing the negative impact on the network when those

resources are temporarily brought down. More frequent network maintenance can in turn significantly increase overall network reliability, thus reducing the likelihood and frequency of unexpected failures within the network.

Having now described several preferred embodiments of the present invention, it should be apparent to those skilled in the art that the foregoing is illustrative only and not limiting, having been presented by way of example only. All the features disclosed in this specification (including any accompanying claims, abstract, and drawings) may be replaced by alternative features serving the same purpose, equivalents or similar purpose, unless expressly stated otherwise. Therefore, numerous other embodiments of the modifications thereof are contemplated as falling within the scope of the present invention as defined by the appended claims and equivalents thereto.

CLAIMS

What is claimed is:

1. In a link state routing protocol network including a plurality of routers having a routing table associated with each of said router and a plurality of links, a method for preventing data loss and forwarding loops during a scheduled network topology change, comprising the ordered steps of:

(a) identifying all affected routers to be changed to effectuate said scheduled network topology change;

(b) performing a predetermined routing table updating sequence for said scheduled network topology change; and

(c) performing said scheduled network topology change.

2. A method as in claim 1 wherein said link state routing protocol is OSPF.

3. A method as in claim 1 wherein the step of identifying all affected routers comprises the steps of:

a. generating a reverse shortest path tree spanning said plurality of routers wherein a router adjacent to said scheduled network topology change is at the root of said reverse shortest path tree; and

b. determining a critical subtree of said reverse shortest path tree that starts at said root and continues along the branch containing said network topology change.

4. A method as in claim 1 wherein said scheduled network topology change is the deletion of a network resource.

5. A method as in claim 4 wherein the step of performing a predetermined routing table sequence comprises updating said routing tables of affected routers farther away from said scheduled network topology change before said routing tables of affected routers nearer to said scheduled network topology.

6. A method as in claim 4 wherein said network resource is chosen from the group consisting of a router and a link.

7. A method as in claim 1 wherein said scheduled network topology change is the addition of a network resource.

8. A method as in claim 7 wherein the step of performing a predetermined routing table sequence further comprises updating said routing tables of affected routers nearer to said scheduled network topology change before said routing tables of affected routers farther away from to said scheduled network topology.

9. A method as in claim 7 wherein said network resource is chosen from the group consisting of a router and a link.

10. In a link state routing protocol network including a plurality of routers having a routing table associated with each of said router and a plurality of links, a computerized system for preventing data loss and forwarding loops during a scheduled network topology change, comprising:

(a) a central processor; and

(b) a program module associated with said central processor for 1) identifying all affected routers to be changed to effectuate said scheduled network topology change; 2) performing a predetermined routing table updating sequence for said scheduled network topology change; and 3) performing said scheduled network topology change.

11. A system as in claim 10 wherein said link state routing protocol is OSPF.

12. A system as in claim 10 wherein the program module, when identifying all said affected routers, generates a reverse shortest path tree spanning said plurality of routers wherein a router adjacent to said scheduled network topology change is at the root of said reverse shortest path tree and determines a critical subtree of said reverse shortest path tree that

starts at said root and continues along the branch containing said network topology change.

13. A system as in claim 10 wherein said scheduled network topology change is the deletion of a network resource.

14. A system as in claim 13 wherein the program module, when performing said predetermined routing table sequence, updates said routing tables of affected routers farther away from said scheduled network topology change before said routing tables of affected routers nearer to said scheduled network topology.

15. A system as in claim 13 wherein said network resource is chosen from the group consisting of a router and a link.

16. A system as in claim 10 wherein said scheduled network topology change is the addition of a network resource.

17. A system as in claim 16 wherein the program module, when performing said predetermined routing table sequence, updates said routing tables of affected routers nearer to said scheduled network topology change before said routing tables of affected routers farther away from to said scheduled network topology.

18. A system as in claim 16 wherein said network resource is chosen from the group consisting of a router and a link.

19. A computer program product for preventing data loss and forwarding loops during a scheduled network topology change to a link state routing protocol network, said network having a plurality of routers having a routing table associated with each of said router and a plurality of links, said computer program product comprising a computer usable medium having computer readable program code embodied in said medium, said program code including: computer readable program code embodied in said computer usable medium for causing a computer to 1) identify all affected routers to be changed to effectuate said

scheduled network topology change; 2) perform a predetermined routing table updating sequence for said scheduled network topology change; and 3) perform said scheduled network topology change.

20. A computer program product as in claim 19 wherein said link state routing protocol is OSPF.

21. A computer program product as in claim 19 wherein said computer readable program code embodied in said computer usable medium program module for causing said computer to identify all said affected routers, causes the computer to generate a reverse shortest path tree spanning said plurality of routers wherein a router adjacent to said scheduled network topology change is at the root of said reverse shortest path tree and determine a critical subtree of said reverse shortest path tree that starts at said root and continues along the branch containing said network topology change.

22. A computer program product as in claim 19 wherein said scheduled network topology change is the deletion of a network resource.

23. A computer program product as in claim 22 wherein said computer readable program code embodied in said computer usable medium program module for causing said computer to perform said predetermined routing table sequence, causes said computer to update said routing tables of affected routers farther away from said scheduled network topology change before said routing tables of affected routers nearer to said scheduled network topology.

24. A computer program product as in claim 22 wherein said network resource is chosen from the group consisting of a router and a link.

25. A computer program product as in claim 19 wherein said scheduled network topology change is the addition of a network resource.

26. A computer program product as in claim 25 wherein said computer readable program code embodied in said computer usable medium program module for causing said computer to perform said predetermined routing table sequence, causes said computer to update said routing tables of affected routers nearer to said scheduled network topology change before said routing tables of affected routers farther away from to said scheduled network topology.

27. A computer program product as in claim 25 wherein said network resource is chosen from the group consisting of a router and a link.

28. In a link state routing protocol network having a network manager, said network including a plurality of routers having a routing table associated with each of said router and a plurality of links, a method, to be performed by said network manager, for preventing data loss and forwarding loops during a scheduled network topology change, comprising the ordered steps of:

(a) identifying all affected routers to be changed to effectuate said network topology change;

(b) performing a predetermined routing table updating sequence; and

(c) performing said scheduled network topology change.

29. A computer program product as in claim 28 wherein said link state routing protocol is OSPF.

30. A computer program product as in claim 28 wherein said scheduled network topology change is the deletion of a network resource.

31. A method as in claim 28 wherein the step of performing said routing table updating sequence further comprises, the steps of:

(a) generating a post change routing table;

(b) comparing said post change routing table with said routing table of each of said affected routers;

(c) updating said routing table of each of said affected routers in accordance with said post change routing table.

32. A computer program product as in claim 28 wherein said network resource is chosen from the group consisting of a router and a link.

33. In a link state routing protocol network including a plurality of routers having a routing table associated with each of said router and a plurality of links, a distributed method for preventing data loss and forwarding loops during a scheduled network topology change, comprising the steps of:

(a) identifying and generating a list of affected routers to be changed to effectuate said network topology change;

(b) generating a plurality of link state forecast messages, each link state forecast message containing said list of affected routers and a forecast of said network topology change;

(c) forwarding said link state forecast messages to each of said affected routers to be acknowledged by each of said affected routers;

(d) generating a post change routing table;

(e) updating said routing table of each of said affected routers in accordance with said post change routing table when all outstanding link state messages have been acknowledged; and

(f) performing said scheduled network topology change.

34. A computer program product as in claim 33 wherein said link state routing protocol is OSPF.

35. A computer program product as in claim 33 wherein said scheduled network topology change is the deletion of a network resource.

36. A computer program product as in claim 33 wherein said network resource is chosen from the group consisting of a router and a link.

FIG. 1A

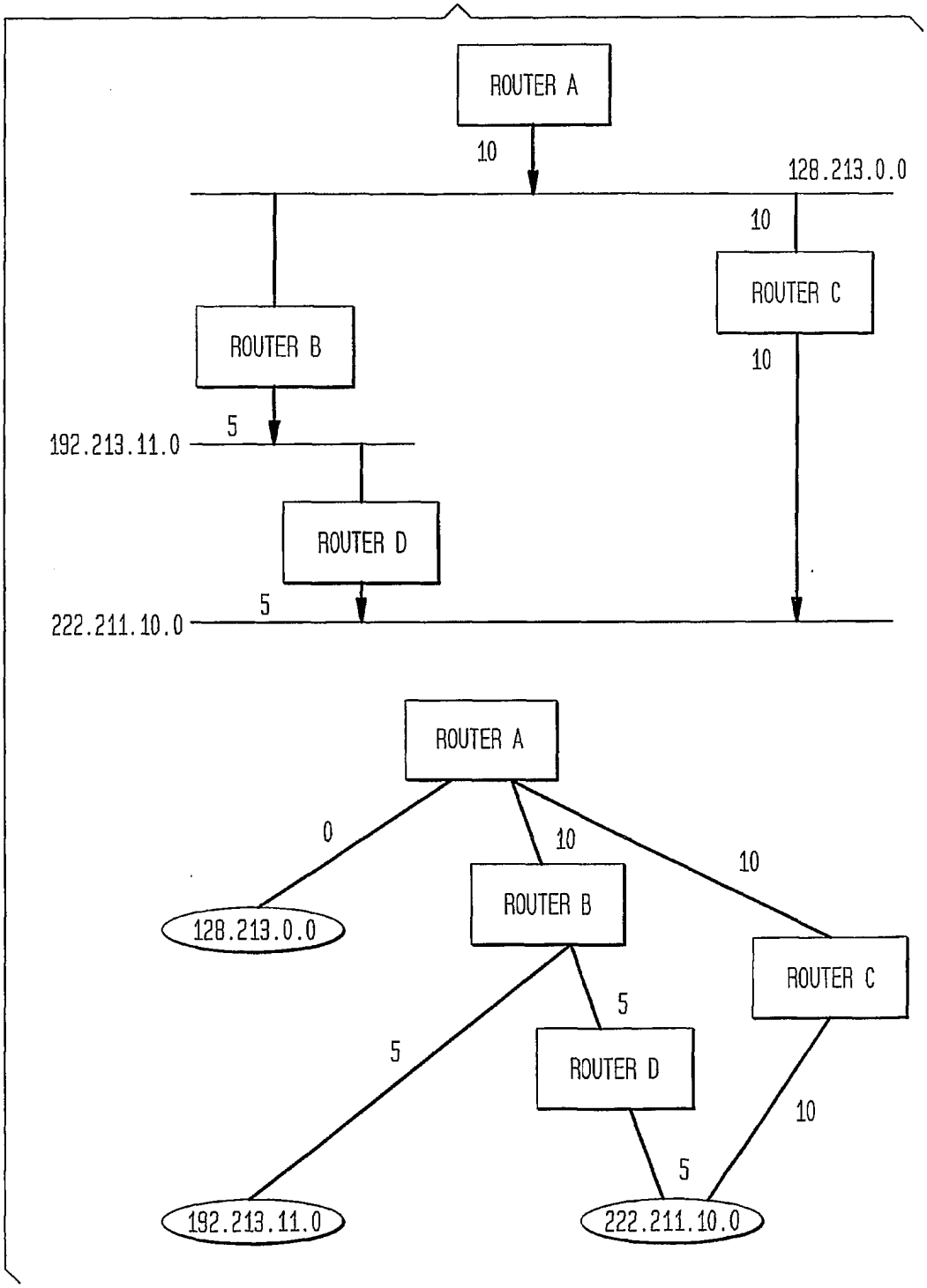

ROUTER A
10
128.213.0.0
10
ROUTER C
ROUTER B
10
5
192.213.11.0
ROUTER D
5
222.211.10.0
ROUTER A
0
10
10
128.213.0.0
ROUTER B
ROUTER C
5
5
ROUTER D
10
5
192.213.11.0
222.211.10.0

FIG. 1B

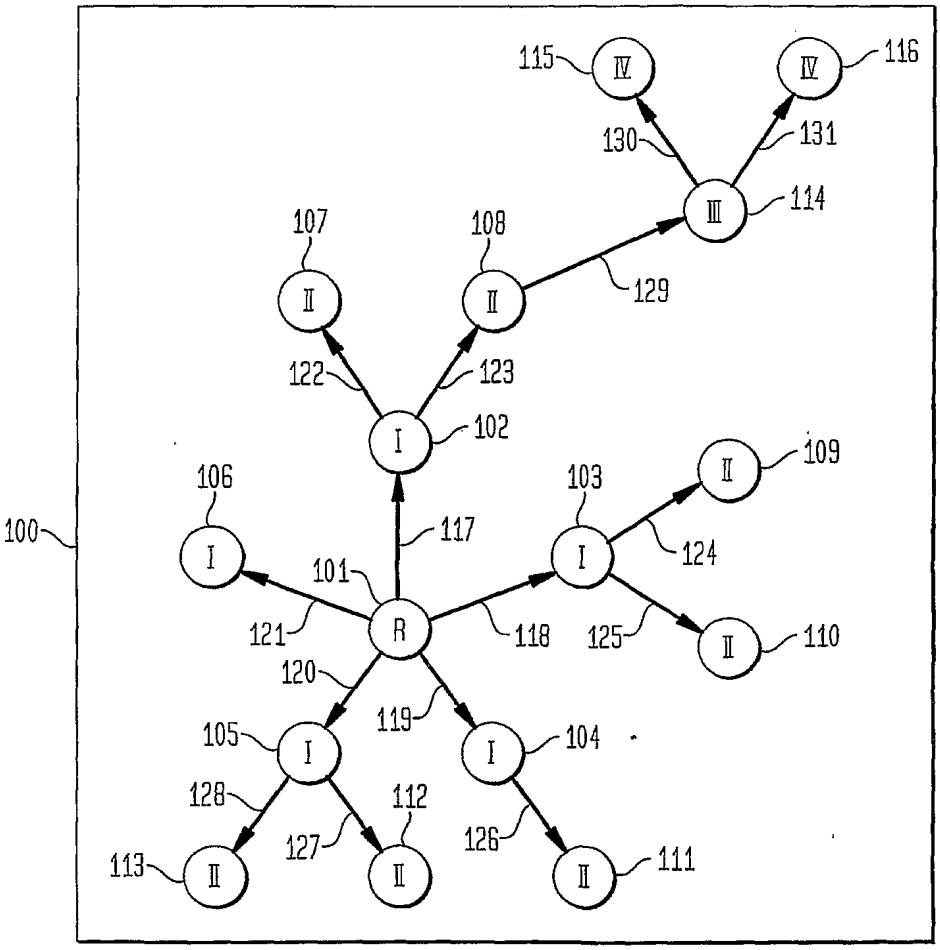

100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
R
I
II
III
IV

FIG. 2

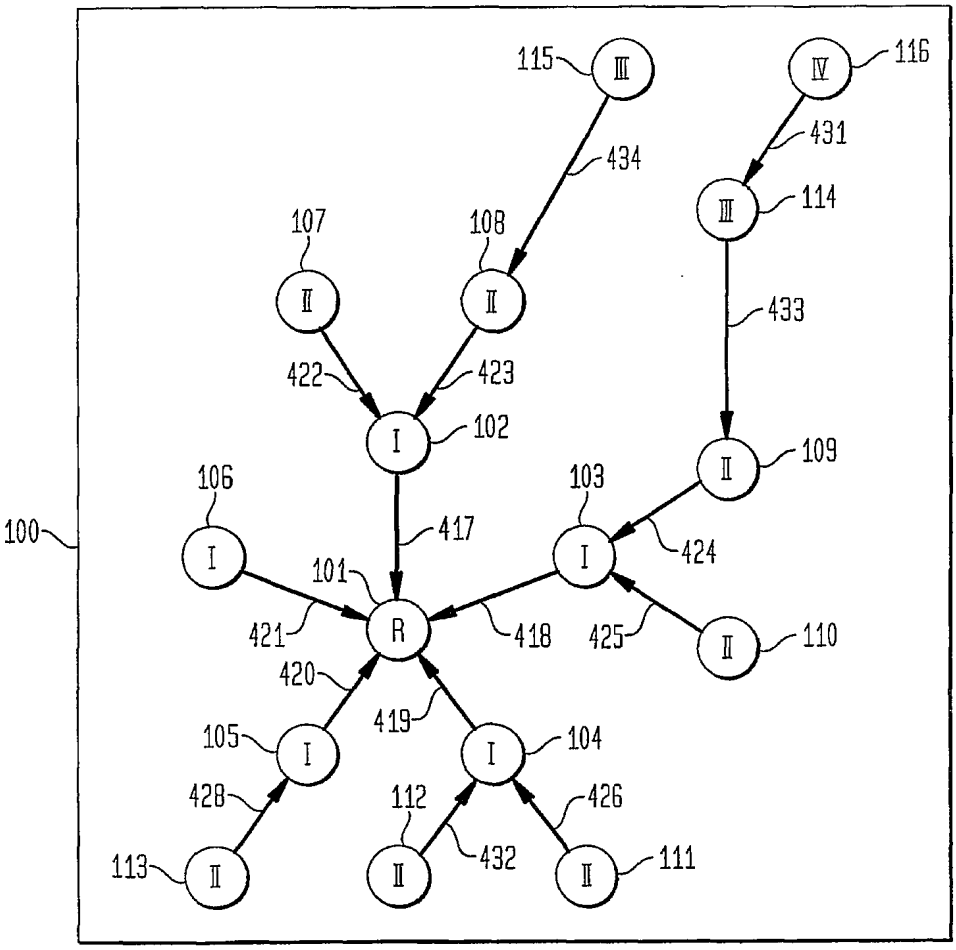

100
115
III
116
IV
431
434
III
114
107
108
II
II
433
422
423
I
102
II
109
106
103
417
I
I
424
101
421
R
418
425
II
110
420
419
105
I
I
104
428
112
426
113
II
II
432
II
111

FIG. 3

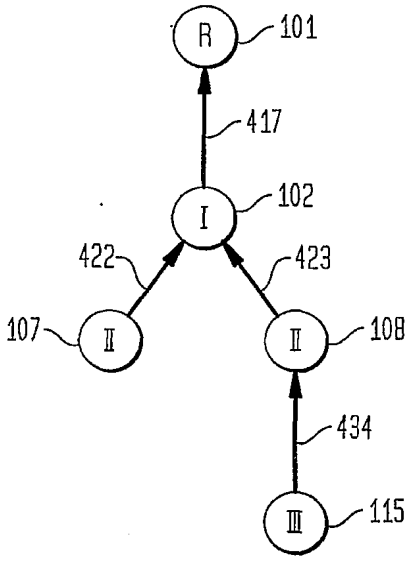

R
101
417
I
102
422
423
107
II
II
108
434
III
115

FIG. 4

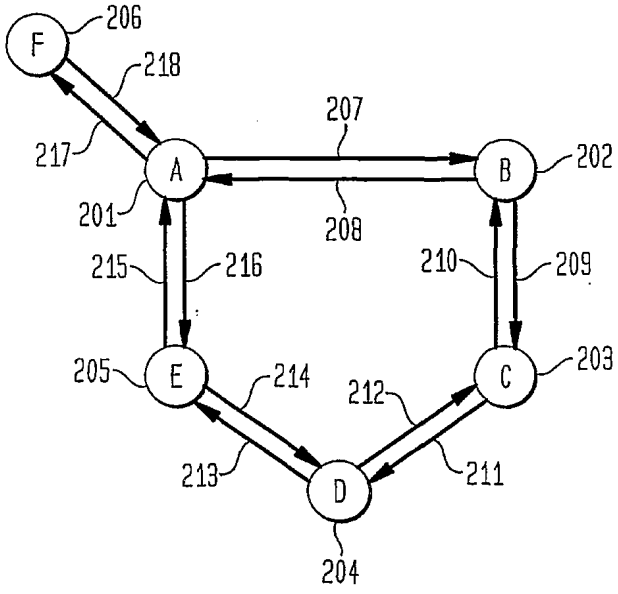

F
206
218
207
217
A
B
202
201
208
215
216
210
209
205
E
214
C
203
212
213
211
D
204

FIG. 5

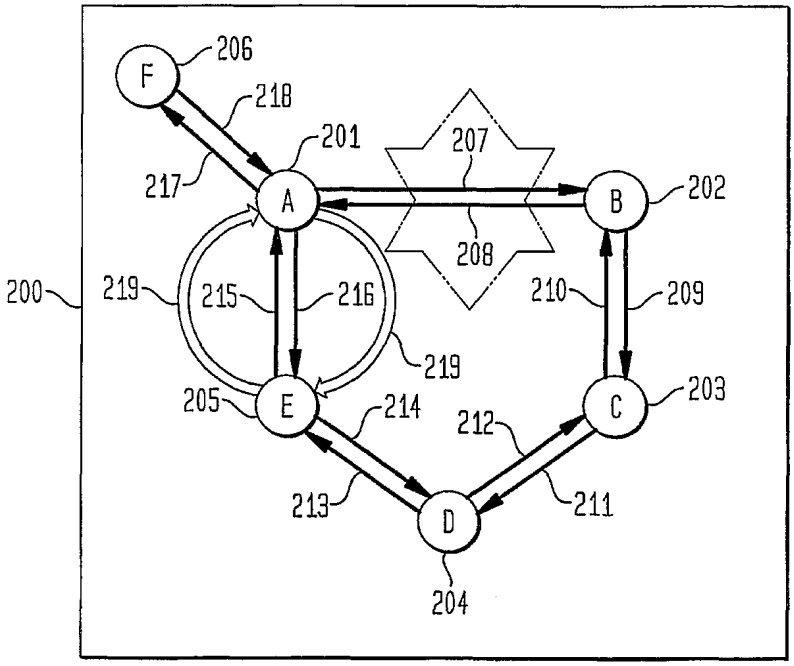

F
206
218
201
207
217
A
B
202
208
200
219
215
216
210
209
219
205
E
214
C
203
212
213
211
D
204

FIG. 6

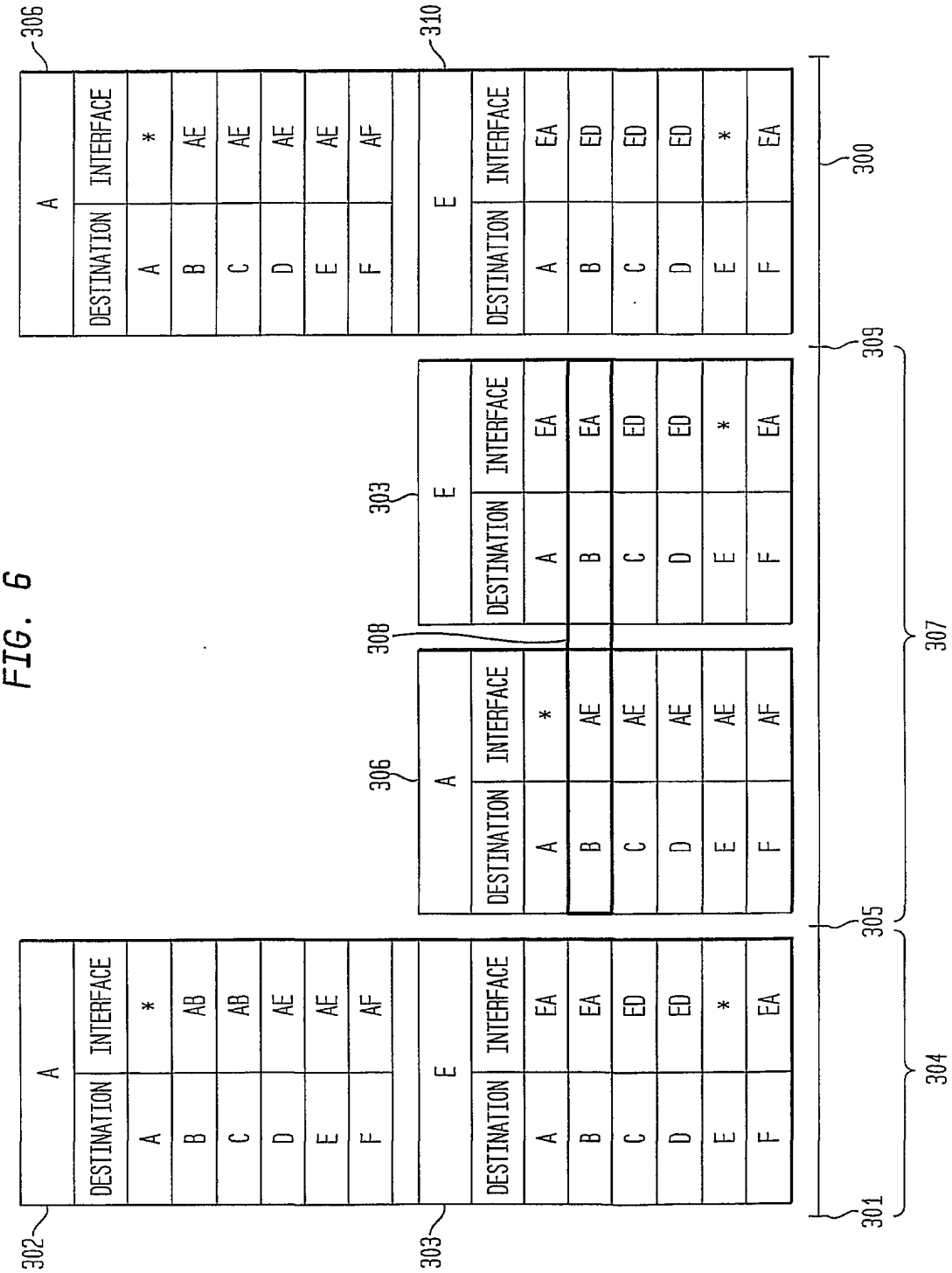

306
A
DESTINATION
INTERFACE
A *
B AE
C AE
D AE
E AE
F AF
310
E
DESTINATION
INTERFACE
A EA
B ED
C ED
D ED
E *
F EA
300
309
307
303
E
DESTINATION
INTERFACE
A EA
B EA
C ED
D ED
E *
F EA
308
306
A
DESTINATION
INTERFACE
A *
B AE
C AE
D AE
E AE
F AF
305
302
A
DESTINATION
INTERFACE
A *
B AB
C AB
D AE
E AE
F AF
303
E
DESTINATION
INTERFACE
A EA
B EA
C ED
D ED
E *
F EA
304
301

FIG. 7

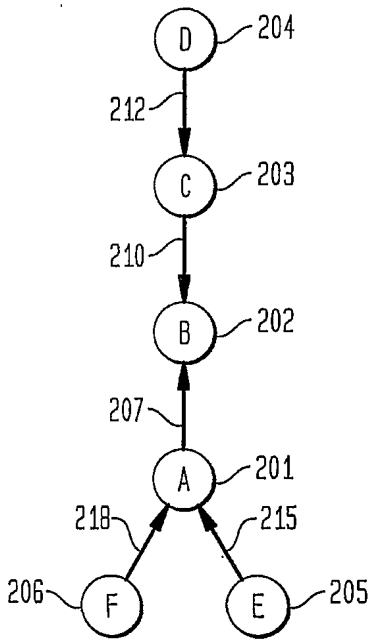

D
204
212
C
203
210
B
202
207
A
201
218
215
206
F
E
205

FIG. 8

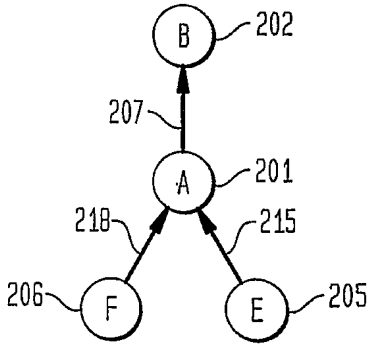

B
202
207
A
201
218
215
206
F
E
205

FIG. 9

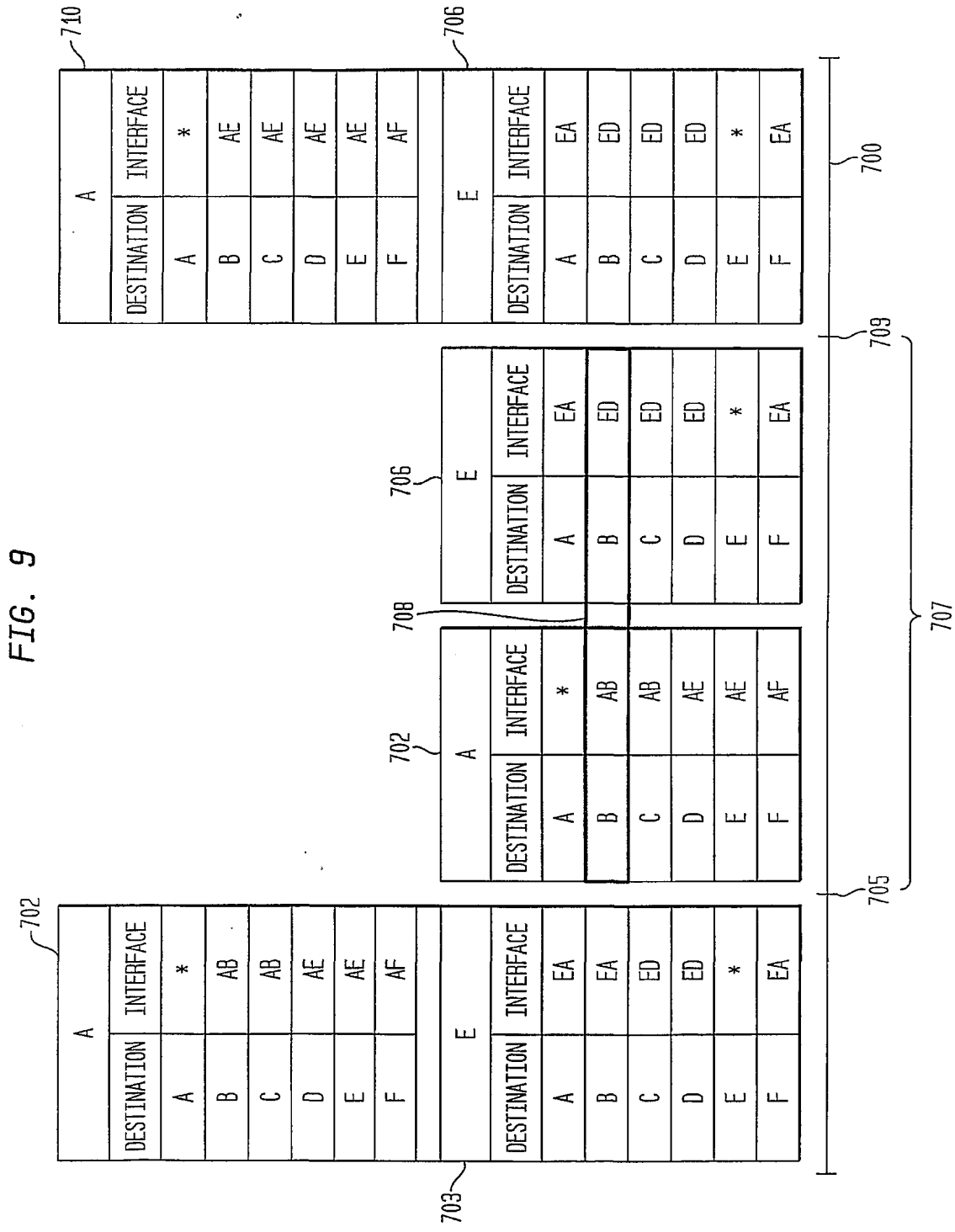

A
DESTINATION
INTERFACE
A *
B AB
C AB
D AE
E AE
F AF
E
DESTINATION
INTERFACE
A EA
B EA
C ED
D ED
E *
F EA
A
DESTINATION
INTERFACE
A *
B AB
C AB
D AE
E AE
F AF
E
DESTINATION
INTERFACE
A EA
B ED
C ED
D ED
E *
F EA
A
DESTINATION
INTERFACE
A *
B AE
C AE
D AE
E AE
F AF
E
DESTINATION
INTERFACE
A EA
B ED
C ED
D ED
E *
F EA
702
703
706
708
702
706
710
705
707
709
700

FIG. 10

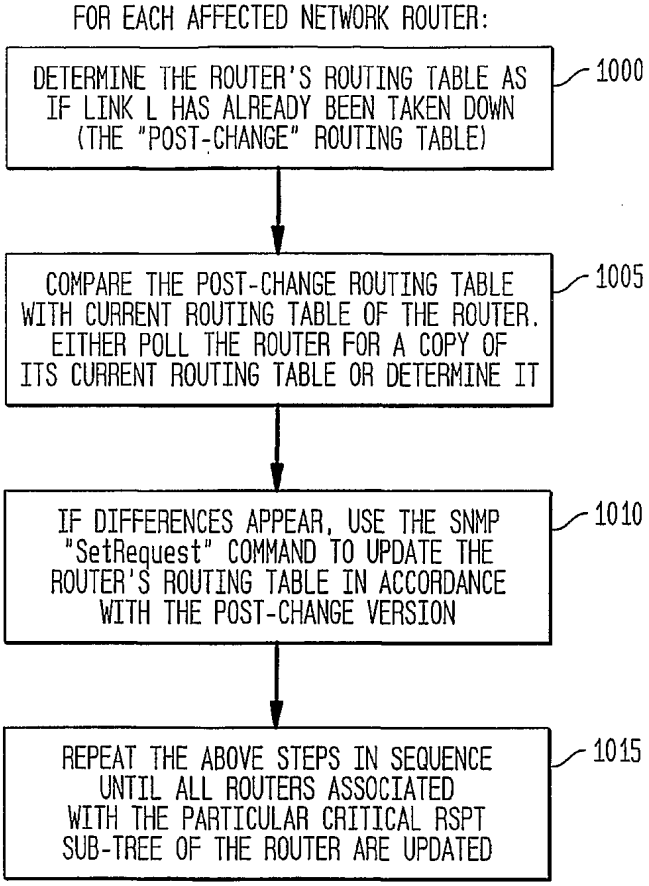
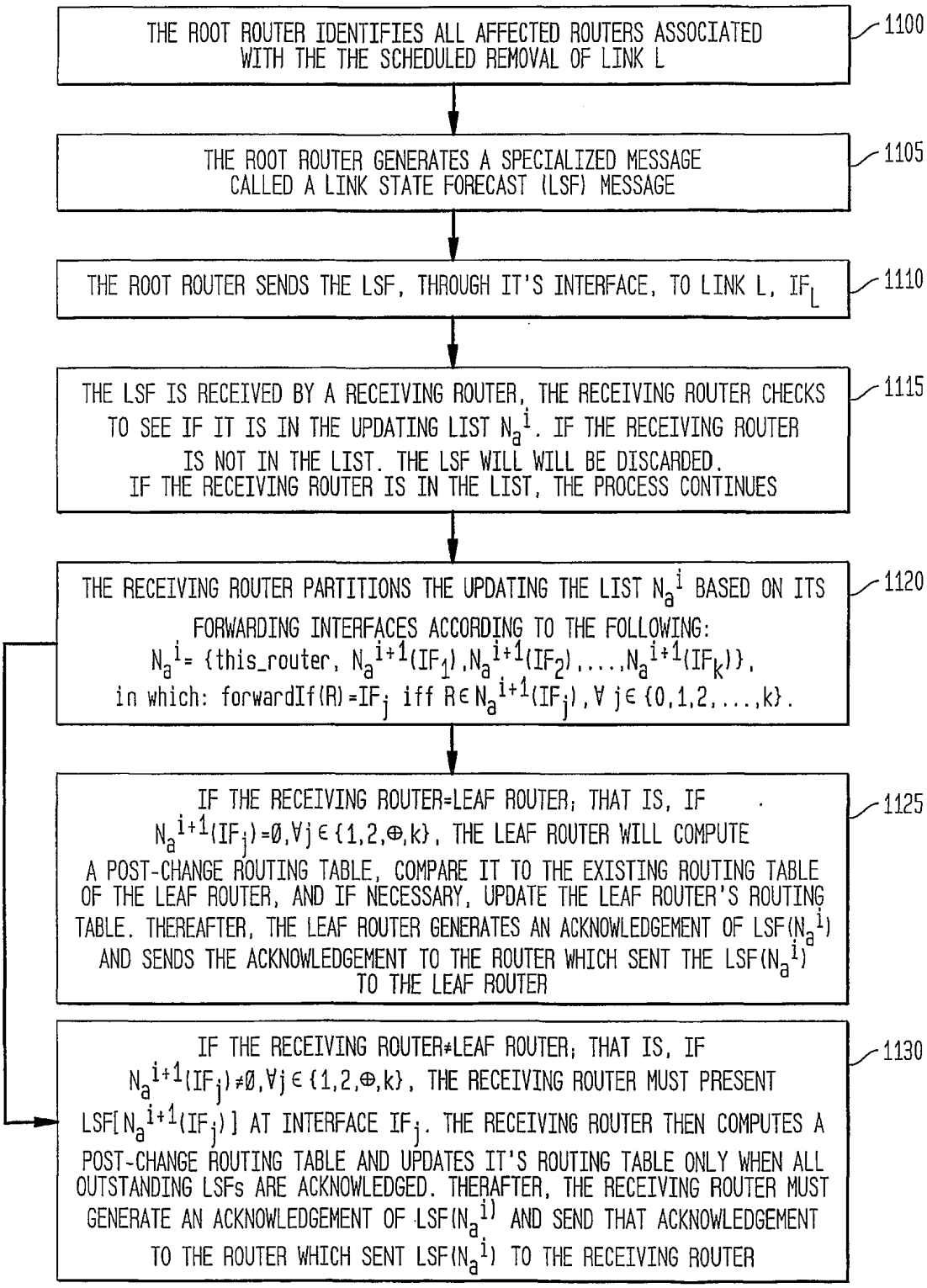

FOR EACH AFFECTED NETWORK ROUTER:
DETERMINE THE ROUTER'S ROUTING TABLE AS IF LINK L HAS ALREADY BEEN TAKEN DOWN (THE "POST-CHANGE" ROUTING TABLE)
1000
COMPARE THE POST-CHANGE ROUTING TABLE WITH CURRENT ROUTING TABLE OF THE ROUTER. EITHER POLL THE ROUTER FOR A COPY OF ITS CURRENT ROUTING TABLE OR DETERMINE IT
1005
IF DIFFERENCES APPEAR, USE THE SNMP "SetRequest" COMMAND TO UPDATE THE ROUTER'S ROUTING TABLE IN ACCORDANCE WITH THE POST-CHANGE VERSION
1010
REPEAT THE ABOVE STEPS IN SEQUENCE UNTIL ALL ROUTERS ASSOCIATED WITH THE PARTICULAR CRITICAL RSPT SUB-TREE OF THE ROUTER ARE UPDATED
1015

FIG. 11


THE ROOT ROUTER IDENTIFIES ALL AFFECTED ROUTERS ASSOCIATED WITH THE THE SCHEDULED REMOVAL OF LINK L
1100
THE ROOT ROUTER GENERATES A SPECIALIZED MESSAGE CALLED A LINK STATE FORECAST (LSF) MESSAGE
1105
THE ROOT ROUTER SENDS THE LSF, THROUGH IT'S INTERFACE, TO LINK L, IF_L
1110
THE LSF IS RECEIVED BY A RECEIVING ROUTER, THE RECEIVING ROUTER CHECKS TO SEE IF IT IS IN THE UPDATING LIST N_a^i. IF THE RECEIVING ROUTER IS NOT IN THE LIST. THE LSF WILL WILL BE DISCARDED. IF THE RECEIVING ROUTER IS IN THE LIST, THE PROCESS CONTINUES
1115
THE RECEIVING ROUTER PARTITIONS THE UPDATING THE LIST N_a^i BASED ON ITS FORWARDING INTERFACES ACCORDING TO THE FOLLOWING:
$N_a^i = \{this\_router, N_a^{i+1}(IF_1), N_a^{i+1}(IF_2), \ldots, N_a^{i+1}(IF_k)\}$,
in which: $forwardIf(R)=IF_j$ iff $R \in N_a^{i+1}(IF_j), \forall j \in \{0,1,2,\ldots,k\}$.
1120
IF THE RECEIVING ROUTER=LEAF ROUTER; THAT IS, IF $N_a^{i+1}(IF_j)=\emptyset, \forall j \in \{1,2,\oplus,k\}$, THE LEAF ROUTER WILL COMPUTE A POST-CHANGE ROUTING TABLE, COMPARE IT TO THE EXISTING ROUTING TABLE OF THE LEAF ROUTER, AND IF NECESSARY, UPDATE THE LEAF ROUTER'S ROUTING TABLE. THEREAFTER, THE LEAF ROUTER GENERATES AN ACKNOWLEDGEMENT OF $LSF(N_a^i)$ AND SENDS THE ACKNOWLEDGEMENT TO THE ROUTER WHICH SENT THE $LSF(N_a^i)$ TO THE LEAF ROUTER
1125
IF THE RECEIVING ROUTER≠LEAF ROUTER; THAT IS, IF $N_a^{i+1}(IF_j)\neq\emptyset, \forall j \in \{1,2,\oplus,k\}$, THE RECEIVING ROUTER MUST PRESENT $LSF[N_a^{i+1}(IF_j)]$ AT INTERFACE IF_j. THE RECEIVING ROUTER THEN COMPUTES A POST-CHANGE ROUTING TABLE AND UPDATES IT'S ROUTING TABLE ONLY WHEN ALL OUTSTANDING LSFs ARE ACKNOWLEDGED. THERAFTER, THE RECEIVING ROUTER MUST GENERATE AN ACKNOWLEDGEMENT OF $LSF(N_a^i)$ AND SEND THAT ACKNOWLEDGEMENT TO THE ROUTER WHICH SENT $LSF(N_a^i)$ TO THE RECEIVING ROUTER
1130